

IN THE CLAIMS:

Please amend the claims as indicated in the complete listing of claims provided below.

1-2. (Canceled)

3. (Currently Amended) A method ~~as in claim 2~~ of managing file extensions in a digital processing system with a user interface and a plurality of files, each file having a name that comprises a filename and an extension, said method comprising:
associating a file with an indicator, the indicator being separate from an extension, the indicator being user selectable for a single file in the plurality of files in said digital processing system without effecting other files in the plurality of files, and the indicator to indicate how to display an extension of the file;
displaying a displayed name of the file in the user interface in a style determined by said indicator; and
wherein the style is such that the displayed name contains the extension of the file only when said indicator is for showing the extension of the file; and
wherein said indicator is a bit, a file, an entry in a file, or an entry in a database,
wherein said indicator in one state indicates hiding the extension and said indicator in another state indicates showing the extension.
4. (Original) A method as in claim 3 wherein if the file is newly created with an automatically appended extension, then said indicator is set to hide the extension of the file in the user interface.

5. (Original) A method as in claim 3 further comprising:
updating said indicator in response to an input event.
6. (Original) A method as in claim 5 wherein the input event is that a new name is
specified in the user interface for the file.
7. (Original) A method as in claim 6 wherein if the new name contains no extension,
then said indicator is set to hide the extension of the file in the user interface.
8. (Original) A method as in claim 7 wherein only the filename of the file is replaced by
the new name so that the extension of the file is not changed.
9. (Original) A method as in claim 6 wherein if the new name contains no extension and
the extension of the file is an empty string, then said indicator is set to a state that
takes a minimum amount of memory to store said state.
10. (Original) A method as in claim 6 wherein if the new name comprising an extension
and a filename, then said indicator is set to show the extension of the file in the user
interface.
11. (Original) A method as in claim 10 wherein the filename of the file and the extension
of the file are replaced by the filename of the new name and the extension of the new
name.

12. (Original) A method as in claim 3 further comprising:
detecting if a first file that has a first filename and a first extension has a naming
conflict with a second file that has a second filename and a second extension,
wherein said first file has a first displayed name in the user interface and said
second file has a second displayed name in the user interface.
13. (Original) A method as in claim 12 wherein if the first displayed name is the same as
the second displayed name, then a naming conflict is detected.
14. (Original) A method as in claim 12 wherein if the first filename and the first
extension are the same as the second filename and the second extension, then a
naming conflict is detected.
15. (Currently Amended) A method as in ~~claim 1~~ claim 3 further comprising:
exporting both the filename of the file and the extension of the file to a remote system
when the file is transferred to the remote system.
16. (Original) A method as in claim 15 further comprising:
exporting said indicator to the remote system when the file is transferred to the remote
system.
17. (Currently Amended) A method as in ~~claim 1~~ claim 3 further comprising:
importing both the filename of the file and the extension of the file from a remote
system when the file is transferred from the remote system.

18. (Original) A method as in claim 17 further comprising:
importing said indicator from the remote system when the file is transferred from the
remote system.

19-23. (Canceled)

24. (Currently Amended) A machine readable medium as in claim 23 for use with a digital processing system which has a user interface and a plurality of files, each file having a name and an extension, said machine readable medium containing executable computer program instructions which when executed by said digital processing system causes said system to perform a method comprising:
associating a file with an indicator, the indicator being separate from an extension, the indicator being user selectable for a single file in the plurality of files in said digital processing system without effecting other files in the plurality of files, and the indicator to indicate how to display an extension of the file; and
displaying a displayed name of the file in the user interface in a style determined by said indicator; and
wherein the style is such that the displayed name contains the extension of the file only when said indicator is for showing the extension of the file; and
wherein said indicator is a bit, a file, an entry in a file, or an entry in a database,
wherein said indicator in one state indicates hiding the extension and said indicator in another state indicates showing the extension.

25. (Previously Presented) A medium as in claim 24 wherein if the file is newly created with an automatically appended extension, then said indicator is set to hide the extension of the file in the user interface.
26. (Previously Presented) A medium as in claim 24 wherein the method further comprises:
updating said indicator in response to an input event.
27. (Previously Presented) A medium as in claim 26 wherein the input event is that a new name is specified in the user interface for the file.
28. (Previously Presented) A medium as in claim 27 wherein if the new name contains no extension, then said indicator is set to hide the extension of the file in the user interface.
29. (Previously Presented) A medium as in claim 28 wherein only the filename of the file is replaced by the new name so that the extension of the file is not changed.
30. (Previously Presented) A medium as in claim 27 wherein if the new name contains no extension and the extension of the file is an empty string, then said indicator is set to a state that takes a minimum amount of memory to store said state.
31. (Previously Presented) A medium as in claim 27 wherein if the new name comprising an extension and a filename, then said indicator is set to show the extension of the file in the user interface.

32. (Previously Presented) A medium as in claim 31 wherein the filename of the file and the extension of the file are replaced by the filename of the new name and the extension of the new name.
33. (Previously Presented) A medium as in claim 24 wherein the method further comprises:
detecting if a first file that has a first filename and a first extension has a naming conflict with a second file that has a second filename and a second extension, wherein said first file has a first displayed name in the user interface and said second file has a second displayed name in the user interface.
34. (Previously Presented) A medium as in claim 33 wherein if the first displayed name is the same as the second displayed name, then a naming conflict is detected.
35. (Previously Presented) A medium as in claim 33 wherein if the first filename and the first extension are the same as the second filename and the second extension, then a naming conflict is detected.
36. (Currently Amended) A medium as in claim ~~[[22]]~~ 24 wherein the method further comprises:
exporting both the filename of the file and the extension of the file to a remote system when the file is transferred to the remote system.
37. (Previously Presented) A medium as in claim 36 wherein the method further comprises:

exporting said indicator to the remote system when the file is transferred to the remote system.

38. (Currently Amended) A medium as in claim [[22]] 24 wherein the method further comprises:

importing both the filename of the file and the extension of the file from a remote system when the file is transferred from the remote system.

39. (Previously Presented) A medium as in claim 38 wherein the method further comprises:

importing said indicator from the remote system when the file is transferred from the remote system.

40-44. (Canceled)

45. (Currently Amended) A ~~system as in claim 44~~ digital processing system with a user interface and a plurality of files, each file having a name that comprises a filename and an extension, said system comprising:

means for associating a file with an indicator, the indicator being separate from an extension, the indicator being user selectable for a single file in the plurality of files in said digital processing system without effecting other files in the plurality of files, and the indicator to indicate how to display extensions of the files; and

means for displaying a displayed name of the file in the user interface in a style determined by said indicator; and

wherein the style is such that the displayed name contains the extension of the file

only when said indicator is for showing the extension of the file; and

wherein said indicator is a bit, a file, an entry in a file, or an entry in a database,

wherein said indicator in one state indicates hiding the extension and said indicator in another state indicates showing the extension.

46. (Original) A system as in claim 45 wherein if the file is newly created with an automatically appended extension, then said indicator is set to hide the extension of the file in the user interface.
47. (Original) A system as in claim 45 further comprising:
means for updating said indicator in response to an input event.
48. (Original) A system as in claim 47 wherein the input event is that a new name is specified in the user interface for the file.
49. (Original) A system as in claim 48 wherein if the new name contains no extension, then said indicator is set to hide the extension of the file in the user interface.
50. (Original) A system as in claim 49 wherein only the filename of the file is replaced by the new name so that the extension of the file is not changed.
51. (Original) A system as in claim 48 wherein if the new name contains no extension and the extension of the file is an empty string, then said indicator is set to a state that takes a minimum amount of memory to store said state.

52. (Original) A system as in claim 48 wherein if the new name comprising an extension and a filename, then said indicator is set to show the extension of the file in the user interface.
53. (Original) A system as in claim 52 wherein the filename of the file and the extension of the file are replaced by the filename of the new name and the extension of the new name.
54. (Original) A system as in claim 45 further comprising:
means for detecting if a first file that has a first filename and a first extension has a naming conflict with a second file that has a second filename and a second extension, wherein said first file has a first displayed name in the user interface and said second file has a second displayed name in the user interface.
55. (Original) A system as in claim 54 wherein if the first displayed name is the same as the second displayed name, then a naming conflict is detected.
56. (Original) A system as in claim 54 wherein if the first filename and the first extension are the same as the second filename and the second extension, then a naming conflict is detected.
57. (Currently Amended) A system as in claim ~~[[43]]~~ 45 further comprising:
means for exporting both the filename of the file and the extension of the file to a remote system when the file is transferred to the remote system.

58. (Original) A system as in claim 57 further comprising:
means for exporting said indicator to the remote system when the file is transferred to
the remote system.
59. (Currently Amended) A system as in claim ~~[[43]]~~ 45 further comprising:
means for importing both the filename of the file and the extension of the file from a
remote system when the file is transferred from the remote system.
60. (Original) A system as in claim 59 further comprising:
means for importing said indicator from the remote system when the file is transferred
from the remote system.
- 61-65. (Canceled)
66. (Currently Amended) A processing system ~~as in claim 65~~ comprising:
a processor;
a display device coupled to said processor, said display device displaying a user
interface; and
a memory coupled to said processor, said memory storing a plurality of files, each file
having a name that comprises a filename and an extension, said memory
storing for a file an indicator, the indicator being separate from an extension,
the indicator being user selectable for a single file in said plurality of files
without effecting other files in the plurality of files, and the indicator to
indicate how to display an extension associated with the file, said processor

displaying a displayed name of said file in said user interface in a style
determined by said indicator; and
wherein the style is such that the displayed name contains the extension of the file
only when said indicator is for showing the extension of the file; and
wherein said indicator is a bit, a file, an entry in a file, or an entry in a database,
wherein said indicator in one state indicates hiding the extension and said
indicator in another state indicates showing the extension.

67. (Original) A processing system as in claim 66 wherein if the file is newly created with an automatically appended extension, then said indicator is set to hide the extension of the file in the user interface.
68. (Original) A processing system as in claim 66 further comprising:
an input device coupled with said processor, said processor updating said indicator in response to an input event detected by said input device.
69. (Original) A processing system as in claim 68 wherein the input event is that a new name is specified in the user interface for the file.
70. (Original) A processing system as in claim 69 wherein if the new name contains no extension, then said indicator is set to hide the extension of the file in the user interface.
71. (Original) A processing system as in claim 70 wherein only the filename of the file is replaced by the new name so that the extension of the file is not changed.

72. (Original) A processing system as in claim 69 wherein if the new name contains no extension and the extension of the file is an empty string, then said indicator is set to a state that takes a minimum amount of memory to store said state.
73. (Original) A processing system as in claim 69 wherein if the new name comprising an extension and a filename, then said indicator is set to show the extension of the file in the user interface.
74. (Original) A processing system as in claim 73 wherein the filename of the file and the extension of the file are replaced by the filename of the new name and the extension of the new name.
75. (Original) A processing system as in claim 66 wherein said processor detects if a first file that has a first filename and a first extension has a naming conflict with a second file that has a second filename and a second extension, wherein said first file has a first displayed name in the user interface and said second file has a second displayed name in the user interface.
76. (Original) A processing system as in claim 75 wherein if the first displayed name is the same as the second displayed name, then a naming conflict is detected.
77. (Original) A processing system as in claim 75 wherein if the first filename and the first extension are the same as the second filename and the second extension, then a naming conflict is detected.

78. (Currently Amended) A processing system as in claim [[64]] 66 further comprising:
a network interface coupled to the said processor, said processor exports both the
filename of the file and the extension of the file to a remote system when the
file is transferred to the remote system through said network interface.
79. (Original) A processing system as in claim 78 wherein said processor exports said
indicator to the remote system when the file is transferred to the remote system
through said network interface.
80. (Currently Amended) A processing system as in claim [[64]] 66 further comprising:
a removable memory coupled to the said processor, said processor exports both the
filename of the file and the extension of the file to a remote system when the
file is transferred to the remote system through said removable memory.
81. (Original) A processing system as in claim 80 wherein said processor exports said
indicator to the remote system when the file is transferred to the remote system
through said removable memory.
82. (Currently Amended) A processing system as in claim [[64]] 66 further comprising:
a network interface coupled to the said processor, said processor imports both the
filename of the file and the extension of the file from a remote system when
the file is transferred from the remote system through said network interface.

83. (Original) A processing system as in claim 82 wherein said processor imports said indicator from the remote system when the file is transferred from the remote system through said network interface.
84. (Currently Amended) A processing system as in claim ~~[[64]]~~ 66 further comprising:
a removable memory coupled to the said processor, said processor imports both the
filename of the file and the extension of the file from a remote system when
the file is transferred from the remote system through said removable memory.
85. (Original) A processing system as in claim 84 wherein said processor imports said indicator from the remote system when the file is transferred from the remote system through said removable memory.

86-107 (Canceled)